**关于核心业务逻辑的观点**

系统作为消息订阅的系统，核心的业务肯定是**通过一个网站的接口获取网站内容的动态变化**，将这种变化及时反馈给用户达到订阅的效果，那么如何获取这个接口，甚至在很多个人网站没有这个接口的情况下如何制作一个接口去监视这个网站，下面我给出几个方案，

Ps:这些方案并不冲突，可以针对不同的网站利用不同的方案去实现，甚至同一个网站可以使用多个方案。

**1.针对性爬虫**

内容：设计针对性很强的爬虫，爬取特定网站内容。

对象：知乎、贴吧、豆瓣等知名度较高需要**细化推送业务**的网站。

优点：单纯的网络爬虫，实现简单。

缺点：只能针对特定网站，每个网站需要设计不同的爬虫逻辑业务。

总结：必不可少的功能模块，设计难度也不大，**解决少量特定网站信息订阅推送问题。**

1. **利用RSS技术**

内容：通过网上流行的RSS技术实现不同网站内容的订阅。

对象：支持RSS技术的网站。

优点：无针对性，所有拥有Feed源的网站均可采用**统一的rss2.0标准**，且实现简单。

缺点：对于不支持RSS技术的网站，只能通过特定不成熟算法或人工操作进行提取Feed。

总结：在已有的技术上进行再加工我认为是非常不错的选择，特别是现在rss越来越被各大网站所接纳，且拥有国际标准，唯一需要解决的就是feed问题。**可以解决部分网站订阅推送问题。**

注意：

对于**不支持rss**技术的网站我们可以使用以下几个方案进行弥补：

(1)**用户自己制作feed**

内容：用户自己进行feed制作（PC端制作），我们提供简易的制作接口（正则匹配之类的接口），用户只需要进行几个简单的操作即可完成feed制作（像feed43那样的制作网站）。

优点：**速度快**，如果制作接口设计的好很容易制作

缺点：用户需要自己挖掘网页源码数据，部分伸手用户可能不愿意制作。

1. **工作人员制作feed**

内容：用户将订阅的网站链接提交至后台，工作人员在24小时内审核并制作完成feed的制作。

优点：方便，解决了一些伸手用户不愿意制作的问题

缺点：加大后台工作人员工作量，可能用户请求得不到及时的反馈

**3.利用算法自动进行Html分析**

内容：对网页源代码进行分析，对每个节点进行权值计算，最终估算网站的树状结构，挖掘网页中我们所需要的信息，制作成我们需要的数据源或者feed。

对象：所有个人或者无feed的网站

优点：自动化，方便，快捷

缺点：算法实现可能不稳定，且目前没有成熟的算法，完全由我们自己造轮子可能有难度。

总结：**可以解决大量网站推送问题**，但实现困难，算是咱们app的核心或者难点。

说明：由于刚想出这个算法，可能漏洞偏大，只是说一下大体思路，日后完善。

网页的html代码是树状的，也就是说，我们要提取的文章列表（这里只以blog首页为标准）肯定是属于某一节点之下的，且结构相同（这里结构相同指的为标签以及class），于是采用以下思想：

1. 对于节点A，如果它的子节点有B,C,D三个节点，那么A的标记为root（这说明A有可能为文章列表根节点），接着往下遍历B,C,D。
2. 对于节点A，如果它的元素存在<h>（标题标签），<p>（文本节点）之类的敏感节点，那么A的权值+Y（敏感权值）
3. 整棵树遍历结束后，权值回推，根据设计的公式（这个需要再考量），计算出的权值最大的节点，可能就是我们的文章列表根节点，子节点的内容，就是我们的文章节点了。
4. 另外，可以建立标签库<article>等标签，比如一些csdn、wordpress等大的博客网站，很多标签标准是统一的，建立标签库可以让我们更方便的匹配到我们需要的内容。
5. 关于个人博客的摘取，个人认为现成的流行博客网站或者框架很有限，大部分个人站也就是套用现成的博客插件比如wordpress制作的，存在统一的标准，所以提取出来感觉不是很麻烦，算法的设计也有可以取巧的地方。